

# Advanced Visualization Moving Rating using Seaborn

```
In [2]: import pandas as pd
```

```
In [3]: movies = pd.read_csv(r"D:\fullstackNaresh\seaborn\Movie-Rating.csv")
```

```
In [4]: movies
```

Out[4]:

	Film	Genre	Rotten Tomatoes Ratings %	Audience Ratings %	Budget (million \$)	Year of release
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009
...	...	...	...	...	...	...
554	Your Highness	Comedy	26	36	50	2011
555	Youth in Revolt	Comedy	68	52	18	2009
556	Zodiac	Thriller	89	73	65	2007
557	Zombieland	Action	90	87	24	2009
558	Zookeeper	Comedy	14	42	80	2011

559 rows × 6 columns

```
In [5]: len(movies)
```

```
Out[5]: 559
```

```
In [6]: import numpy as np
```

```
In [7]: movies.columns
```

```
Out[7]: Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
       'Budget (million $)', 'Year of release'],
       dtype='object')
```

```
In [8]: movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Film              559 non-null    object  
 1   Genre             559 non-null    object  
 2   Rotten Tomatoes Ratings %  559 non-null    int64  
 3   Audience Ratings %  559 non-null    int64  
 4   Budget (million $) 559 non-null    int64  
 5   Year of release   559 non-null    int64  
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

In [9]: `movies.shape`

Out[9]: (559, 6)

In [10]: `movies.head()`

Out[10]:

	Film	Genre	Rotten Tomatoes Ratings %	Audience Ratings %	Budget (million \$)	Year of release
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009

In [11]: `movies.tail()`

Out[11]:

	Film	Genre	Rotten Tomatoes Ratings %	Audience Ratings %	Budget (million \$)	Year of release
554	Your Highness	Comedy	26	36	50	2011
555	Youth in Revolt	Comedy	68	52	18	2009
556	Zodiac	Thriller	89	73	65	2007
557	Zombieland	Action	90	87	24	2009
558	Zookeeper	Comedy	14	42	80	2011

In [12]: `movies.describe()`

Out[12]:

	Rotten Tomatoes Ratings %	Audience Ratings %	Budget (million \$)	Year of release
<b>count</b>	559.000000	559.000000	559.000000	559.000000
<b>mean</b>	47.309481	58.744186	50.236136	2009.152057
<b>std</b>	26.413091	16.826887	48.731817	1.362632
<b>min</b>	0.000000	0.000000	0.000000	2007.000000
<b>25%</b>	25.000000	47.000000	20.000000	2008.000000
<b>50%</b>	46.000000	58.000000	35.000000	2009.000000
<b>75%</b>	70.000000	72.000000	65.000000	2010.000000
<b>max</b>	97.000000	96.000000	300.000000	2011.000000

In [14]: `movies.columns = ['Film', 'Genre', 'CriticRating', 'AudienceRating', 'BudgetMillions', '`In [15]: `movies.head(0)`Out[15]: `Film Genre CriticRating AudienceRating BudgetMillions Year`In [16]: `movies.describe()`

	CriticRating	AudienceRating	BudgetMillions	Year
<b>count</b>	559.000000	559.000000	559.000000	559.000000
<b>mean</b>	47.309481	58.744186	50.236136	2009.152057
<b>std</b>	26.413091	16.826887	48.731817	1.362632
<b>min</b>	0.000000	0.000000	0.000000	2007.000000
<b>25%</b>	25.000000	47.000000	20.000000	2008.000000
<b>50%</b>	46.000000	58.000000	35.000000	2009.000000
<b>75%</b>	70.000000	72.000000	65.000000	2010.000000
<b>max</b>	97.000000	96.000000	300.000000	2011.000000

In [19]: `movies.Film = movies.Film.astype('category')`In [20]: `movies.describe()`

Out[20]:

	CriticRating	AudienceRating	BudgetMillions	Year
<b>count</b>	559.000000	559.000000	559.000000	559.000000
<b>mean</b>	47.309481	58.744186	50.236136	2009.152057
<b>std</b>	26.413091	16.826887	48.731817	1.362632
<b>min</b>	0.000000	0.000000	0.000000	2007.000000
<b>25%</b>	25.000000	47.000000	20.000000	2008.000000
<b>50%</b>	46.000000	58.000000	35.000000	2009.000000
<b>75%</b>	70.000000	72.000000	65.000000	2010.000000
<b>max</b>	97.000000	96.000000	300.000000	2011.000000

In [22]:

```
movies.Year = movies.Year.astype('category')
movies.Genre = movies.Genre.astype('category')
```

In [23]:

```
movies.describe()
```

Out[23]:

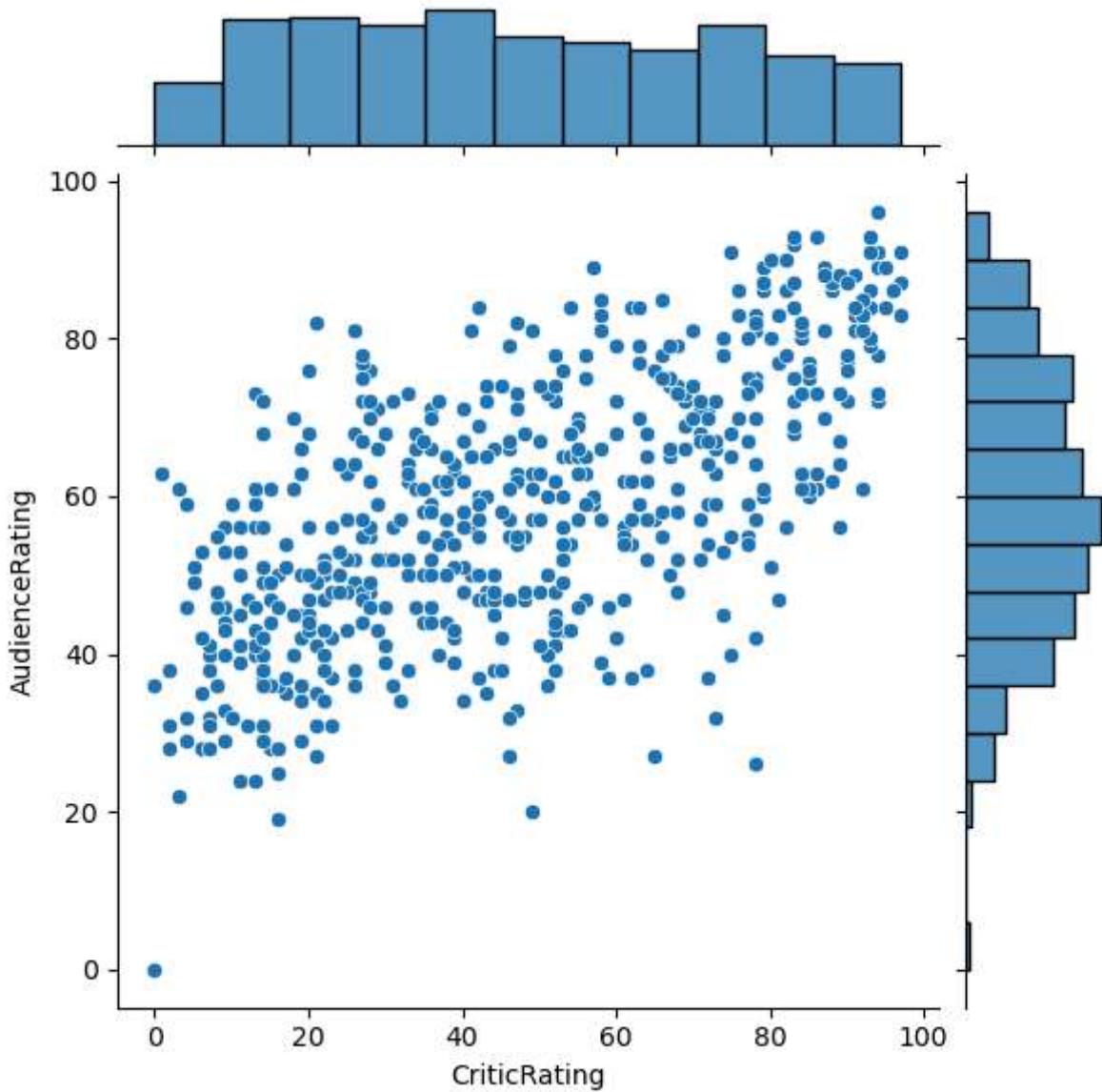
	CriticRating	AudienceRating	BudgetMillions
<b>count</b>	559.000000	559.000000	559.000000
<b>mean</b>	47.309481	58.744186	50.236136
<b>std</b>	26.413091	16.826887	48.731817
<b>min</b>	0.000000	0.000000	0.000000
<b>25%</b>	25.000000	47.000000	20.000000
<b>50%</b>	46.000000	58.000000	35.000000
<b>75%</b>	70.000000	72.000000	65.000000
<b>max</b>	97.000000	96.000000	300.000000

In [32]:

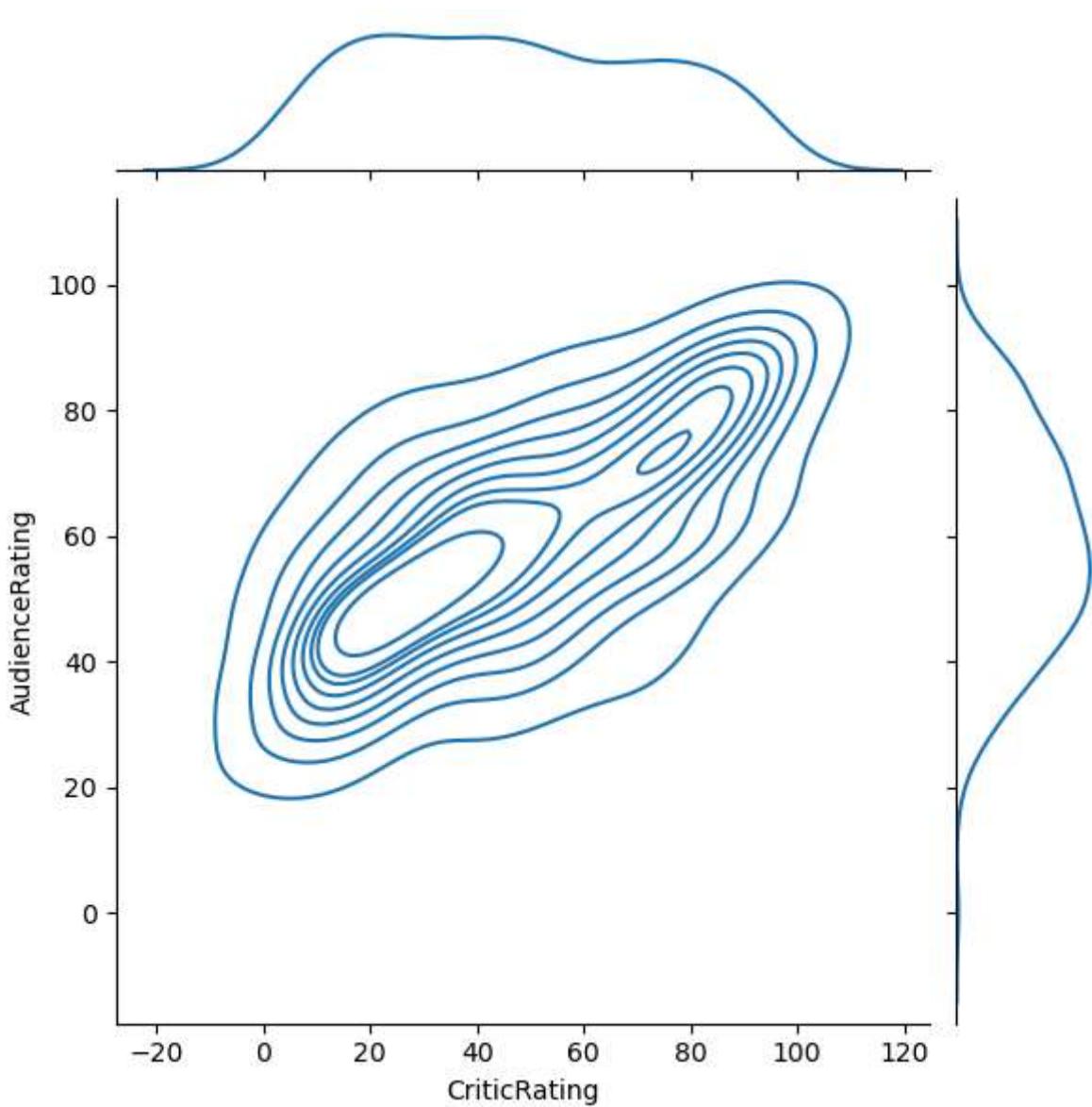
```
#How to working joint plots
from matplotlib import pyplot as plt
import seaborn as sns

%matplotlib inline

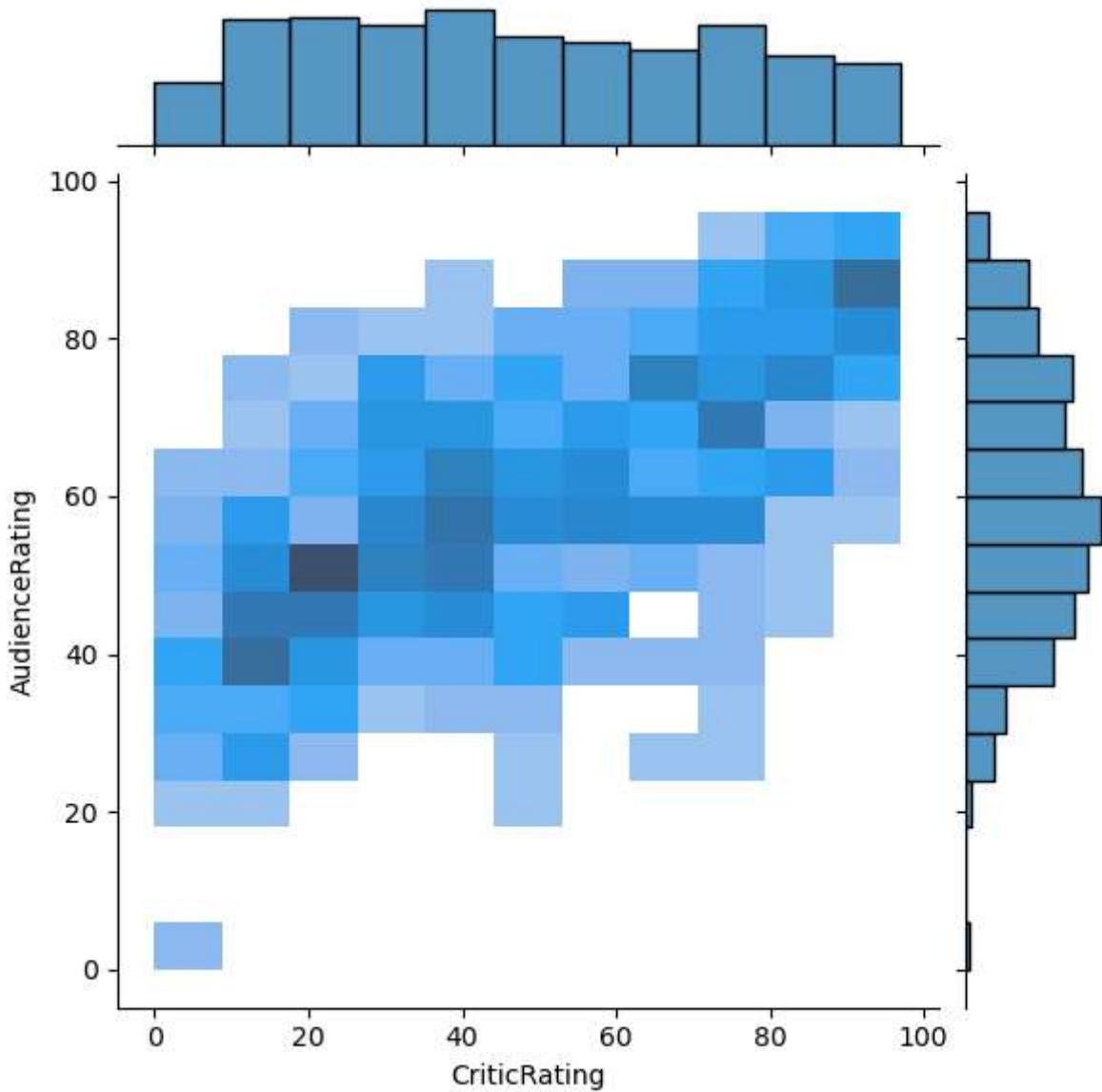
import warnings
warnings.filterwarnings("ignore")
j = sns.jointplot(data = movies, x ='CriticRating' , y ='AudienceRating',kind='scat
plt.show()
```



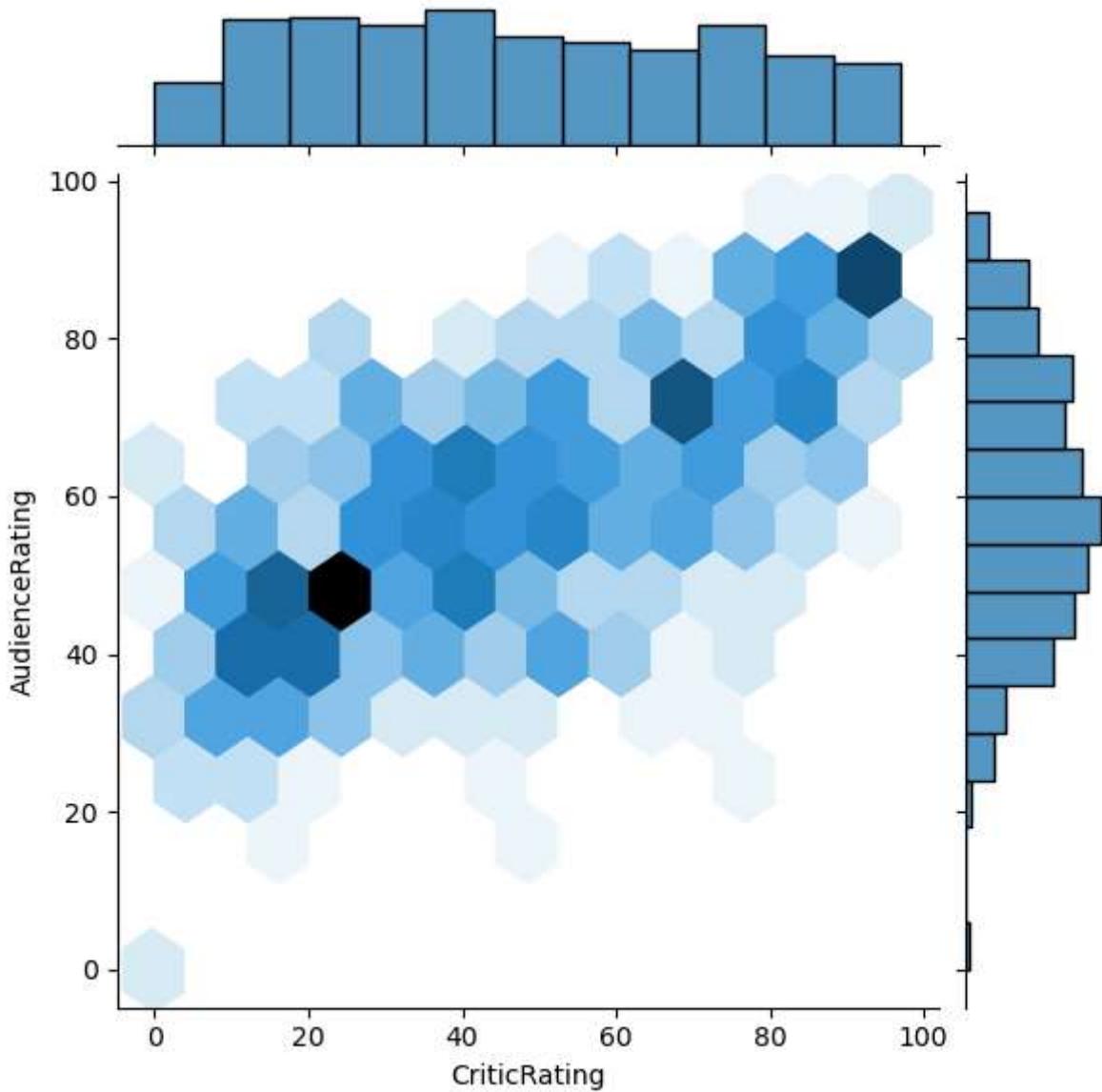
```
In [36]: j = sns.jointplot(data = movies, x ='CriticRating' , y ='AudienceRating',kind='kde'  
plt.show()
```



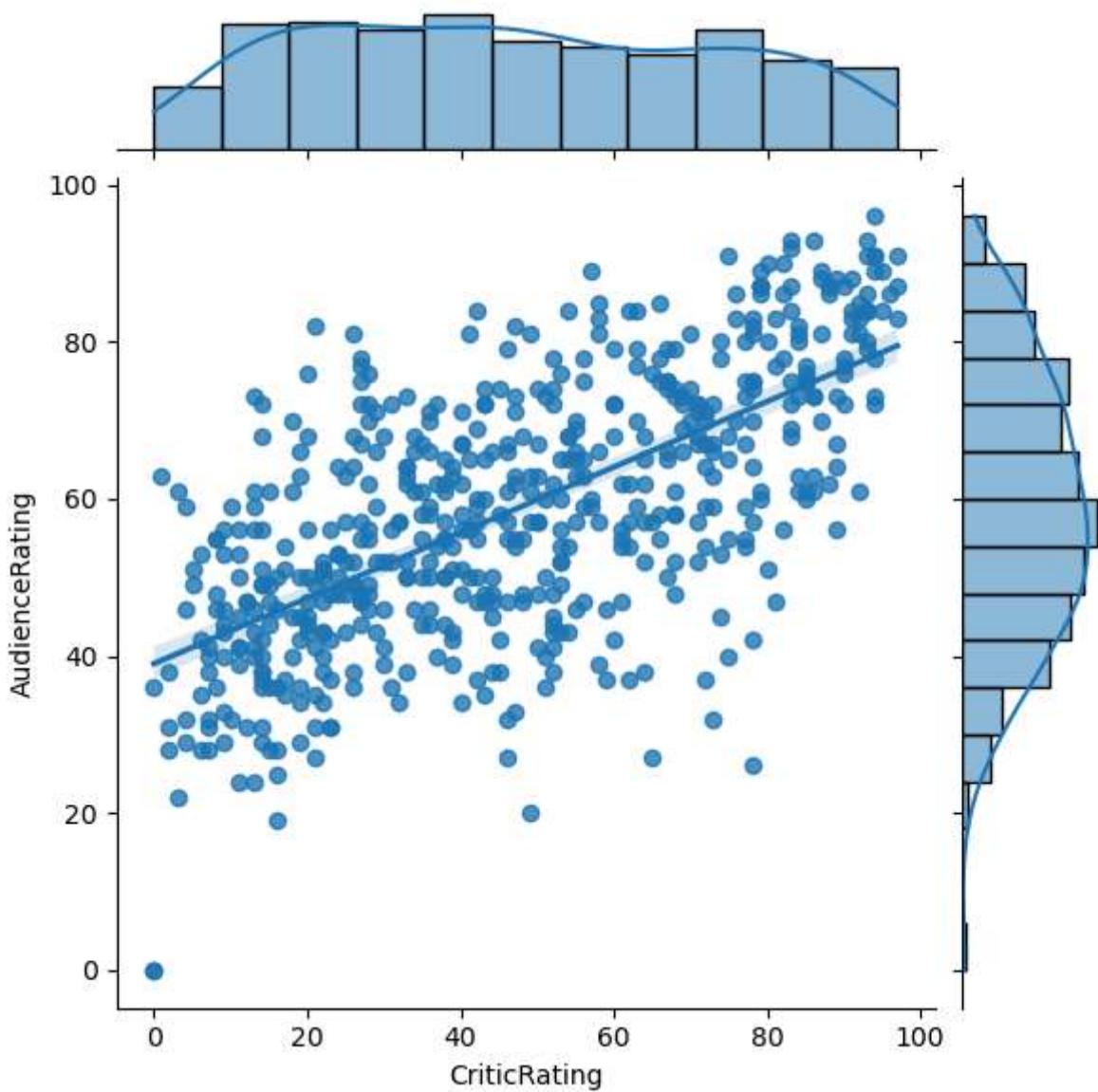
```
In [37]: j = sns.jointplot(data = movies, x ='CriticRating' , y ='AudienceRating',kind='hist')
plt.show()
```



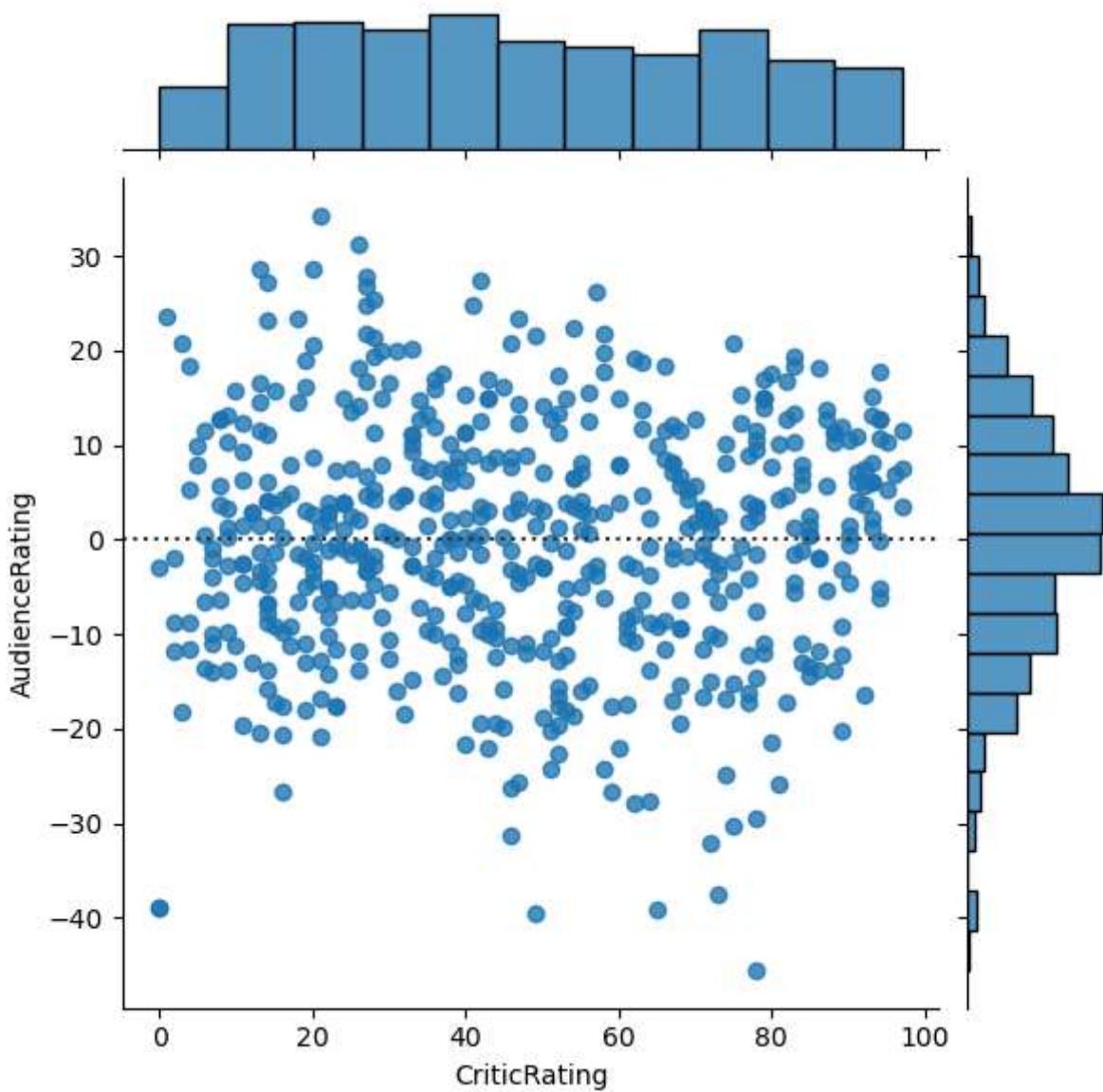
```
In [38]: j = sns.jointplot(data = movies, x ='CriticRating' , y ='AudienceRating',kind='hex'  
plt.show()
```



```
In [39]: j = sns.jointplot(data = movies, x ='CriticRating' , y ='AudienceRating',kind='reg'  
plt.show()
```

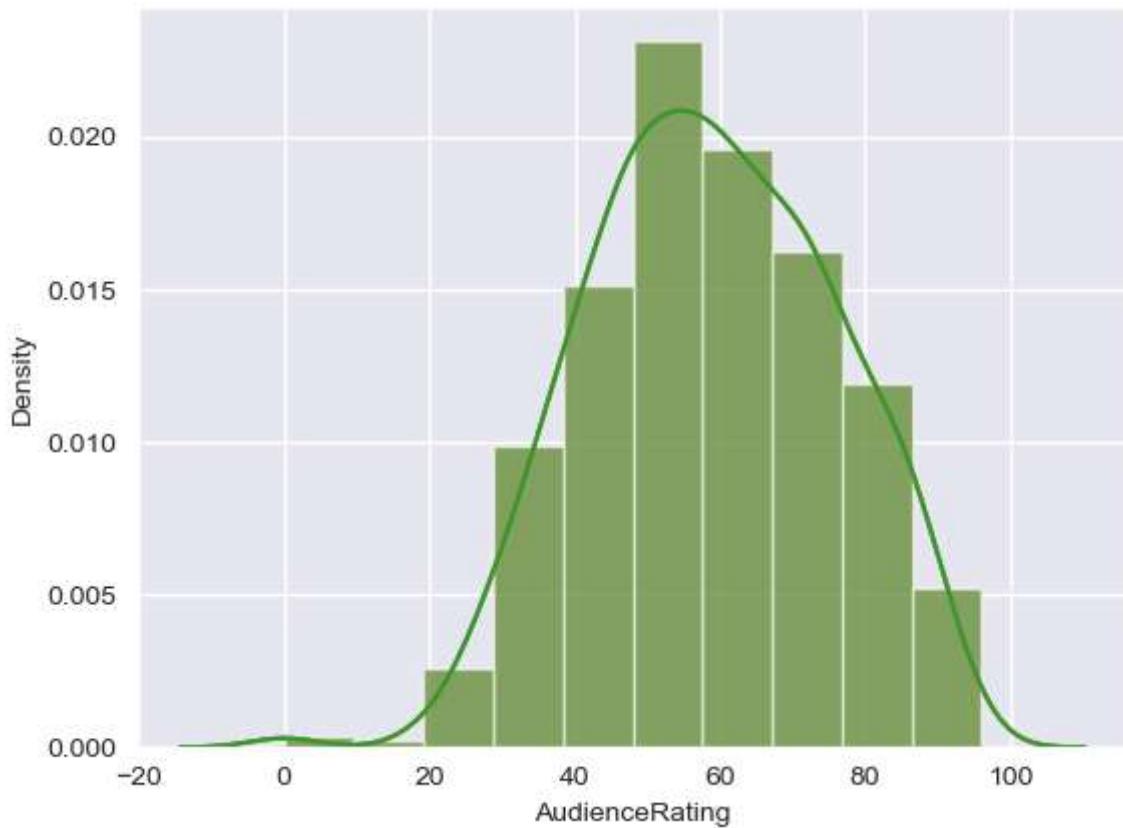


```
In [40]: j = sns.jointplot(data = movies, x ='CriticRating' , y ='AudienceRating',kind='resid')
plt.show()
```

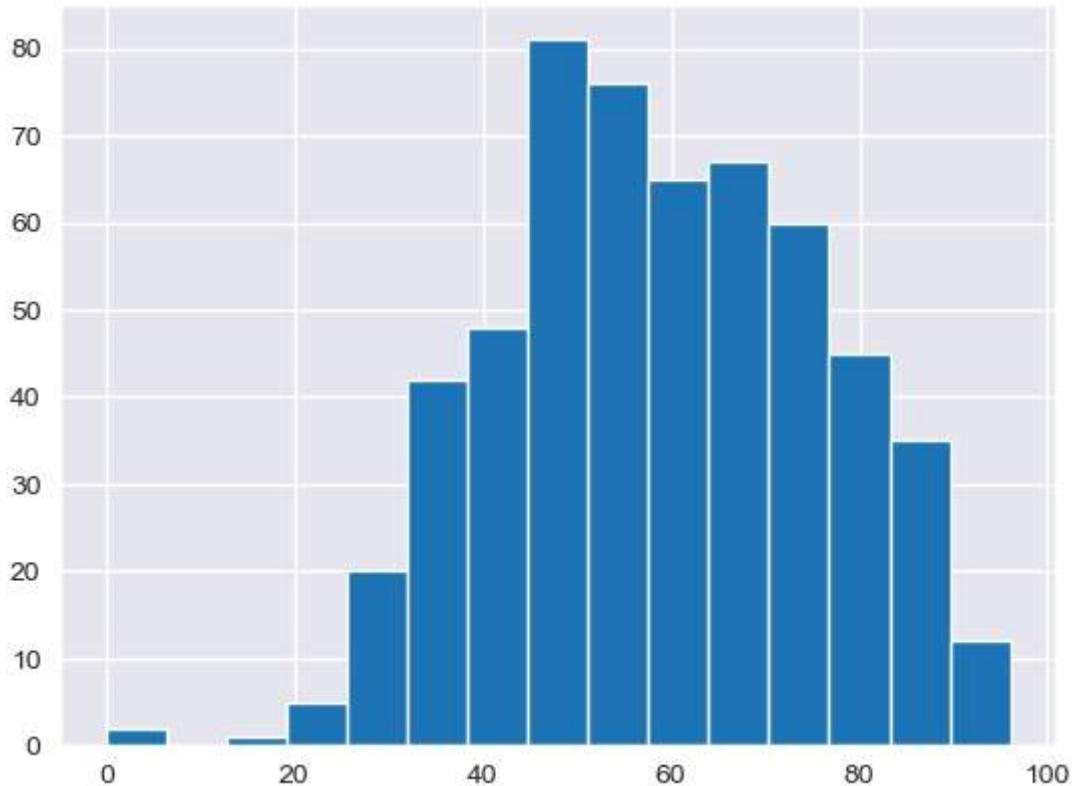


```
In [41]: sns.set_style('darkgrid')
```

```
In [45]: m2 = sns.distplot(movies.AudienceRating,bins=10)  
plt.show()
```

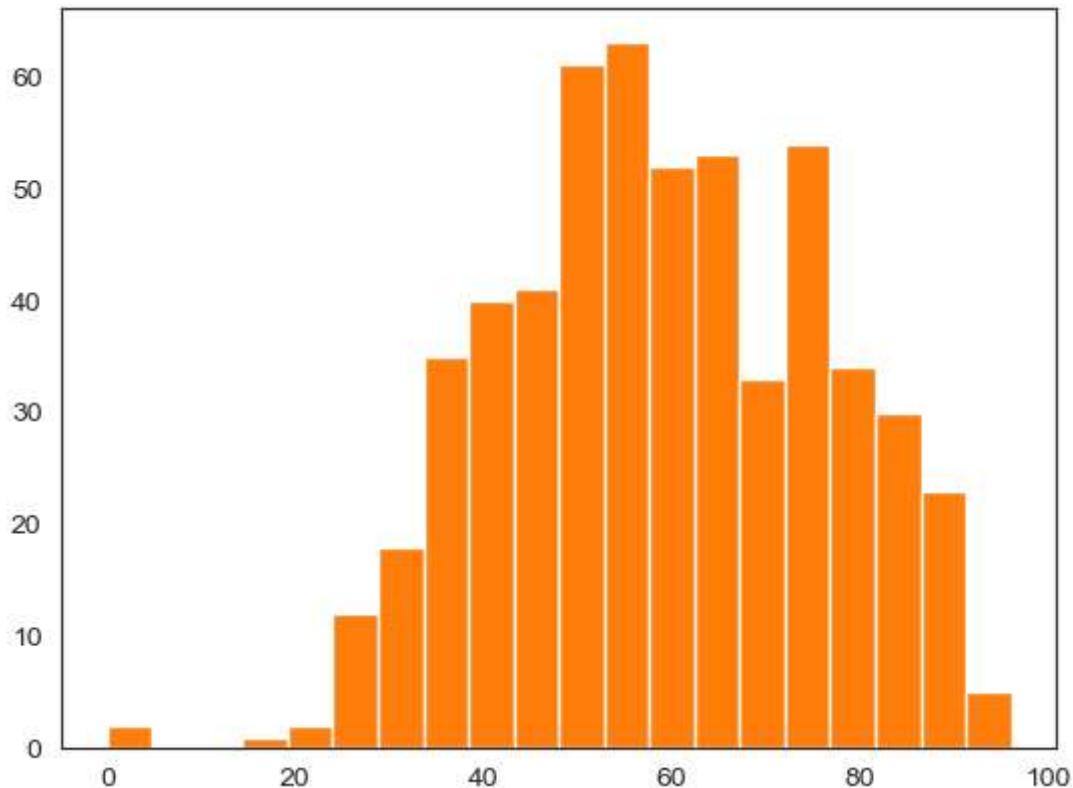


```
In [46]: plt.hist(movies.AudienceRating, bins=15)  
plt.show()
```

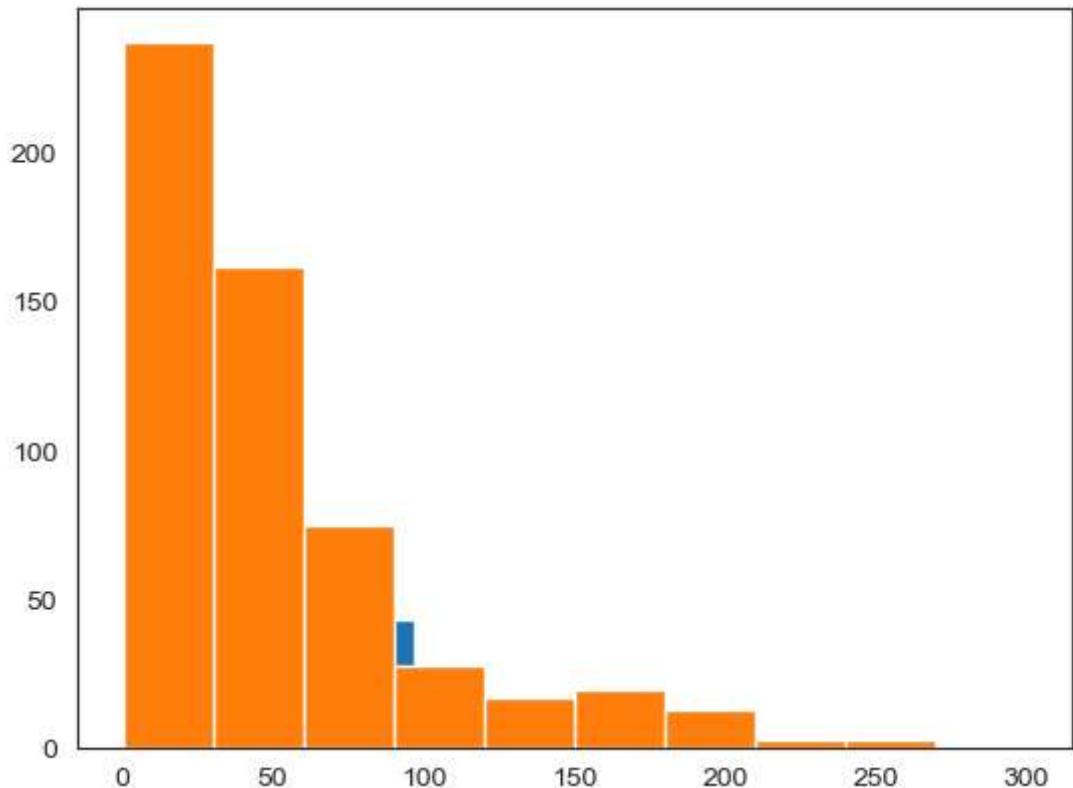


```
In [48]: sns.set_style('white')  
n1=plt.hist(movies.AudienceRating, bins=20)
```

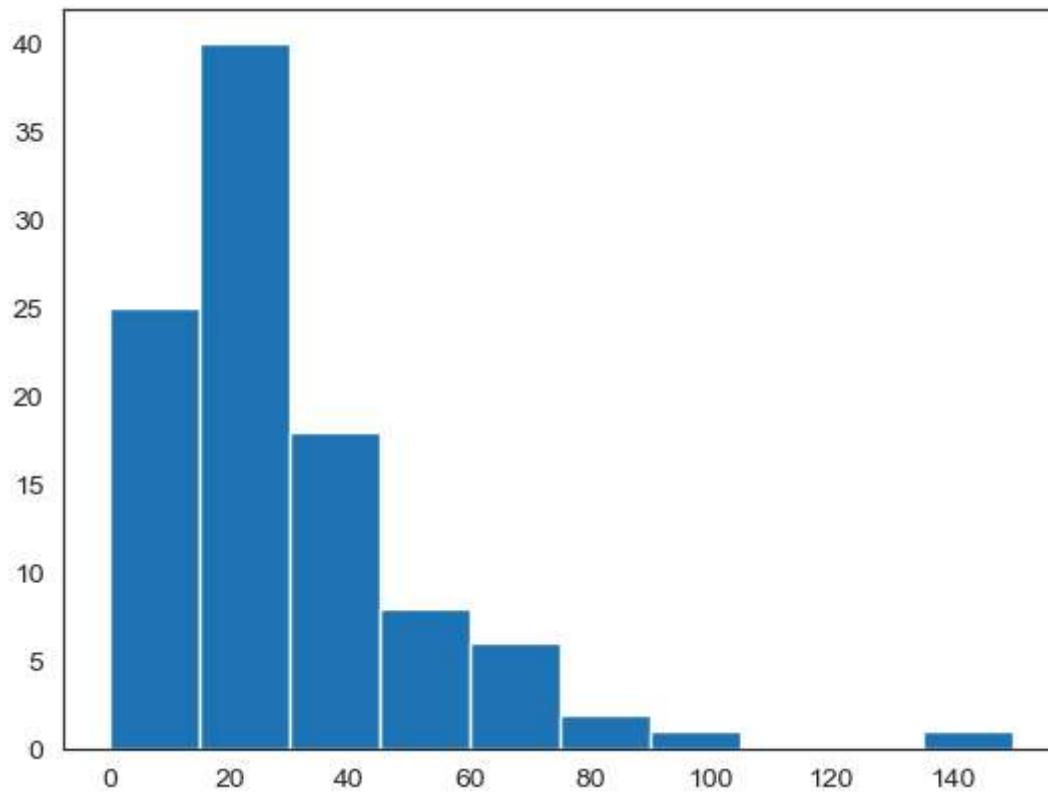
```
plt.show()
```



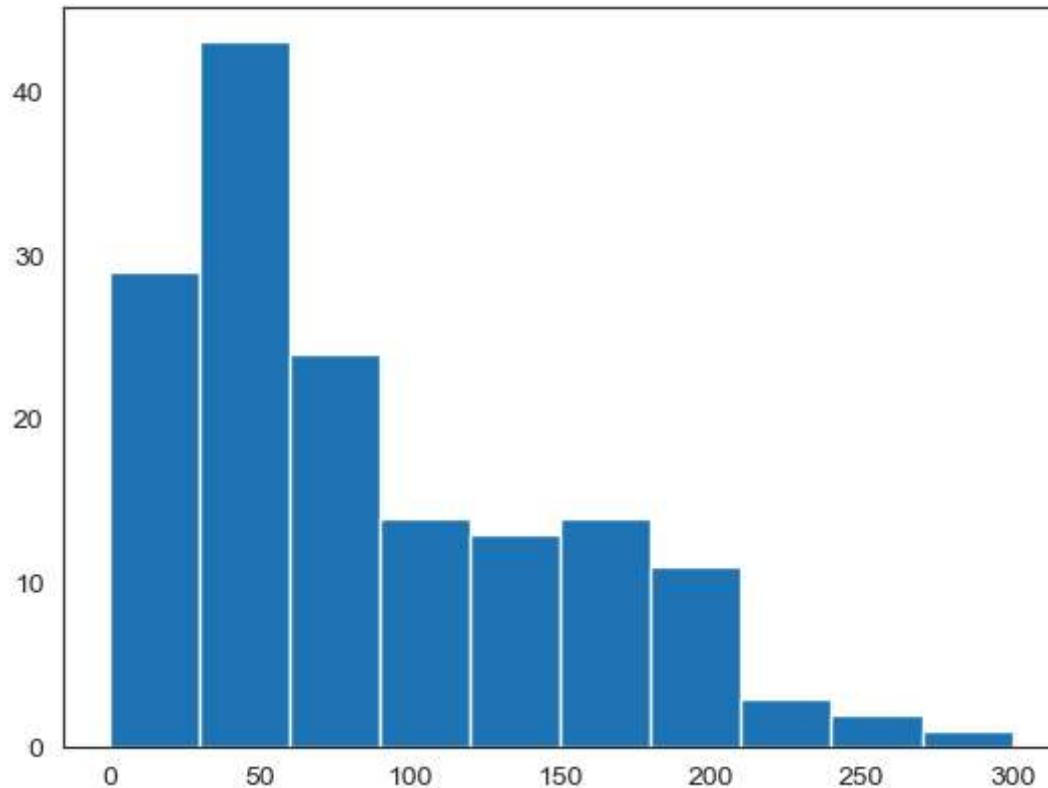
```
In [52]: h1 = plt.hist(movies.BudgetMillions)  
plt.show()
```



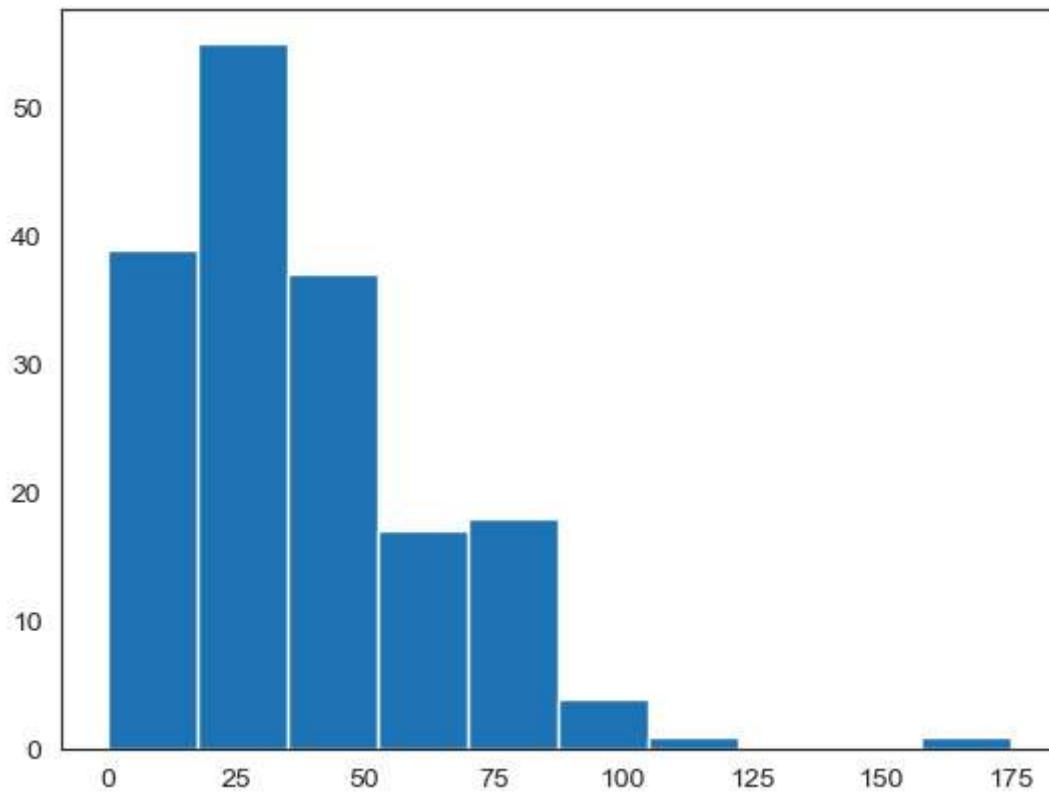
```
In [55]: plt.hist(movies[movies.Genre=='Drama'].BudgetMillions)  
plt.show()
```



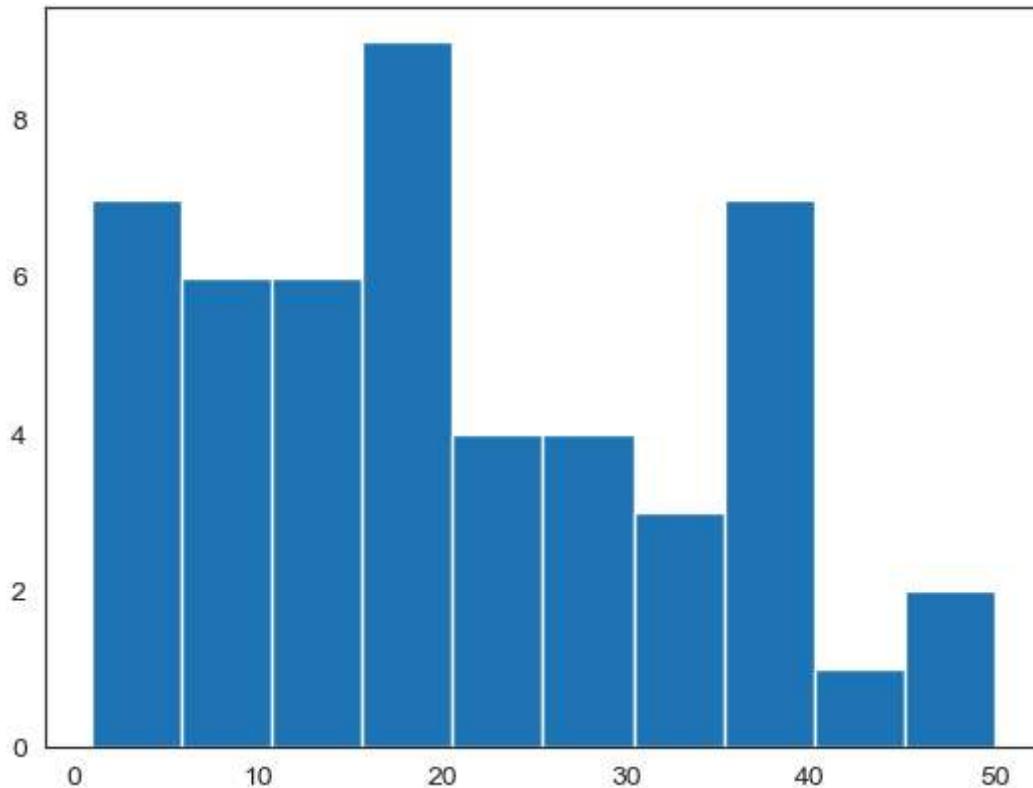
```
In [56]: plt.hist(movies[movies.Genre == 'Action'].BudgetMillions)  
plt.show()
```



```
In [57]: plt.hist(movies[movies.Genre == 'Comedy'].BudgetMillions)
plt.show()
```



```
In [58]: plt.hist(movies[movies.Genre == 'Horror'].BudgetMillions)
plt.show()
```



In [60]: `movies.head()`

Out[60]:

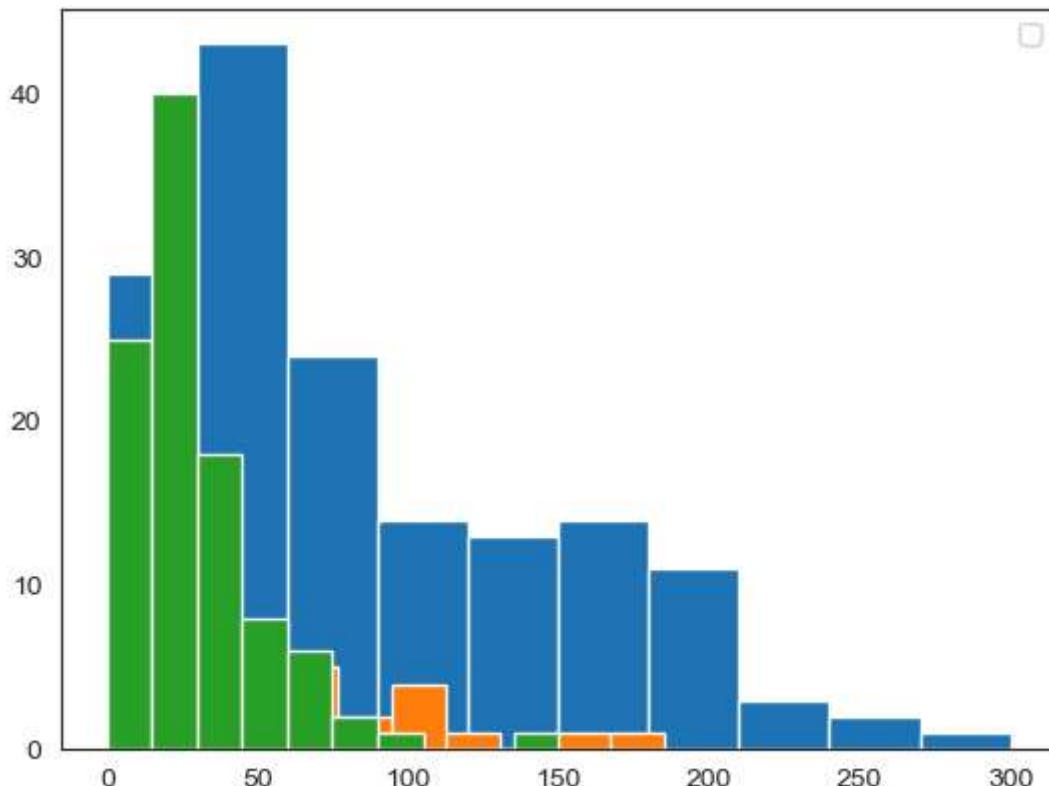
	Film	Genre	CriticRating	AudienceRating	BudgetMillions	Year
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009

In [61]: `movies.Genre.unique()`

Out[61]: `['Comedy', 'Adventure', 'Action', 'Horror', 'Drama', 'Romance', 'Thriller']`  
`Categories (7, object): ['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance', 'Thriller']`

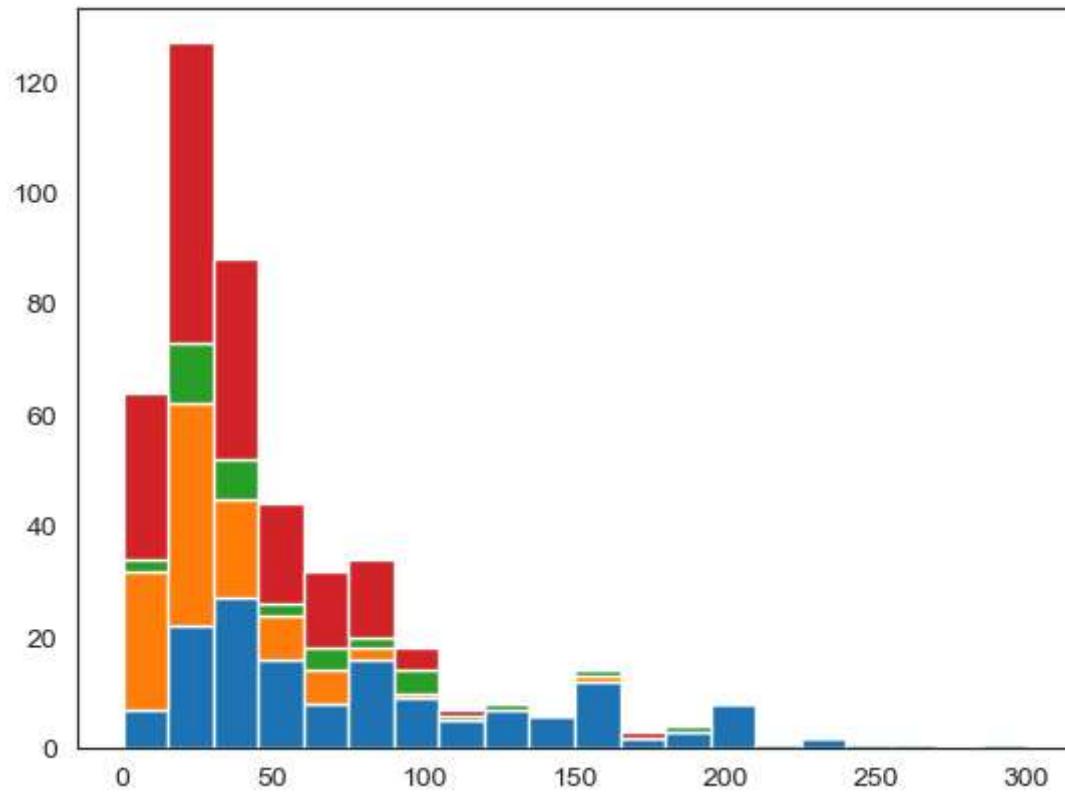
In [62]: `# Stacked hist`

```
plt.hist(movies[movies.Genre == 'Action'].BudgetMillions)
plt.hist(movies[movies.Genre == 'Thriller'].BudgetMillions)
plt.hist(movies[movies.Genre == 'Drama'].BudgetMillions)
plt.legend()
plt.show()
```



In [65]: `plt.hist([movies[movies.Genre == 'Action'].BudgetMillions,\n movies[movies.Genre == 'Drama'].BudgetMillions,\n movies[movies.Genre == 'Thriller'].BudgetMillions,\n movies[movies.Genre == 'Action'].BudgetMillions,\n movies[movies.Genre == 'Drama'].BudgetMillions,\n movies[movies.Genre == 'Thriller'].BudgetMillions],\n bins=10, stacked=True)`

```
movies[movies.Genre == 'Comedy'].BudgetMillions,  
bins = 20 ,stacked = True)  
plt.show()
```



```
In [68]: for gen in movies.Genre.cat.categories:  
    print(gen)
```

Action  
Adventure  
Comedy  
Drama  
Horror  
Romance  
Thriller

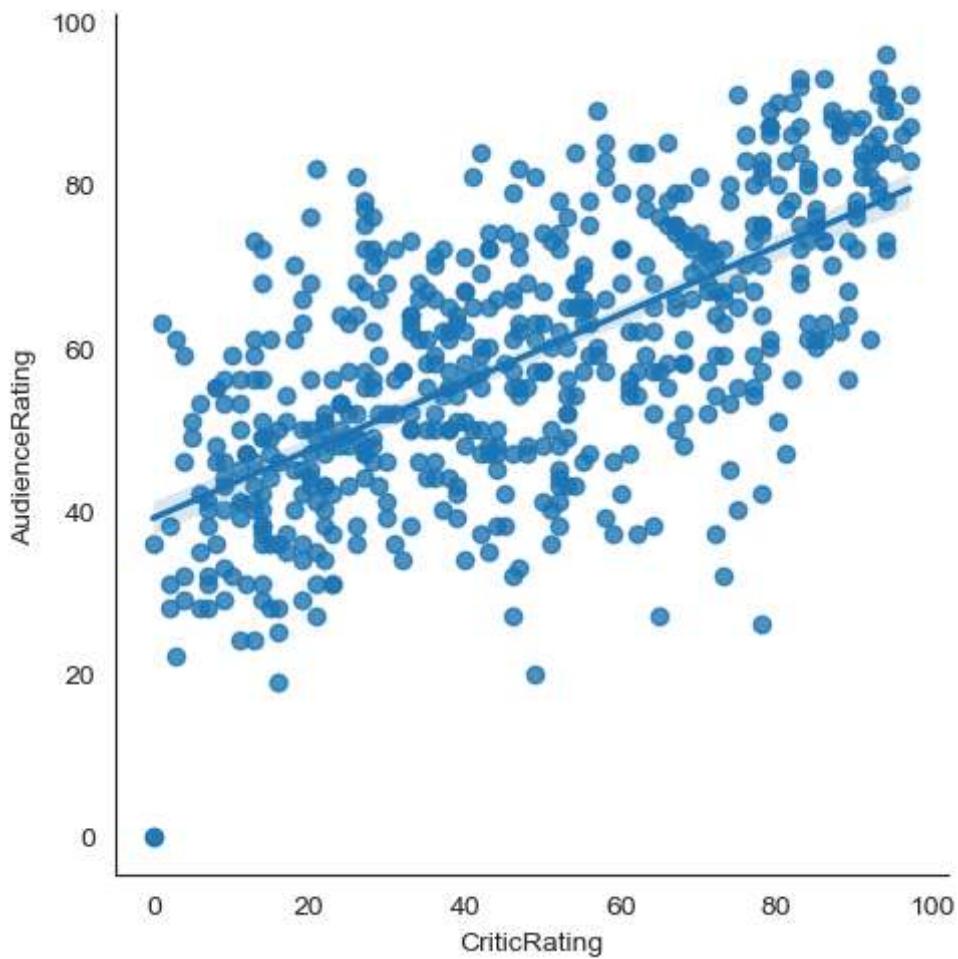
```
In [78]: data =movies  
data
```

Out[78]:

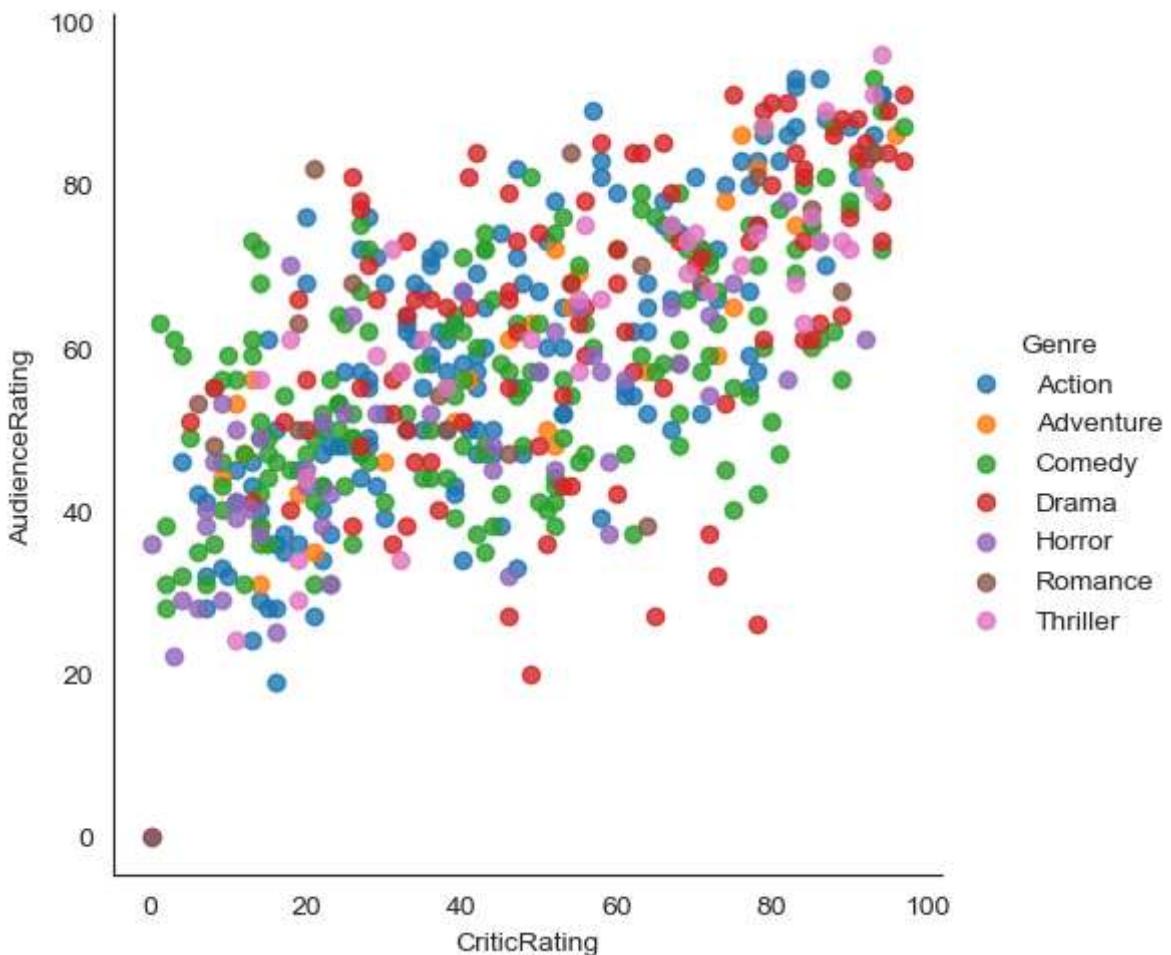
	Film	Genre	CriticRating	AudienceRating	BudgetMillions	Year
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009
...	...	...	...	...	...	...
554	Your Highness	Comedy	26	36	50	2011
555	Youth in Revolt	Comedy	68	52	18	2009
556	Zodiac	Thriller	89	73	65	2007
557	Zombieland	Action	90	87	24	2009
558	Zookeeper	Comedy	14	42	80	2011

559 rows × 6 columns

In [82]: `vis1 = sns.lmplot(data=data, x='CriticRating',y='AudienceRating',fit_reg=True)  
plt.show()`

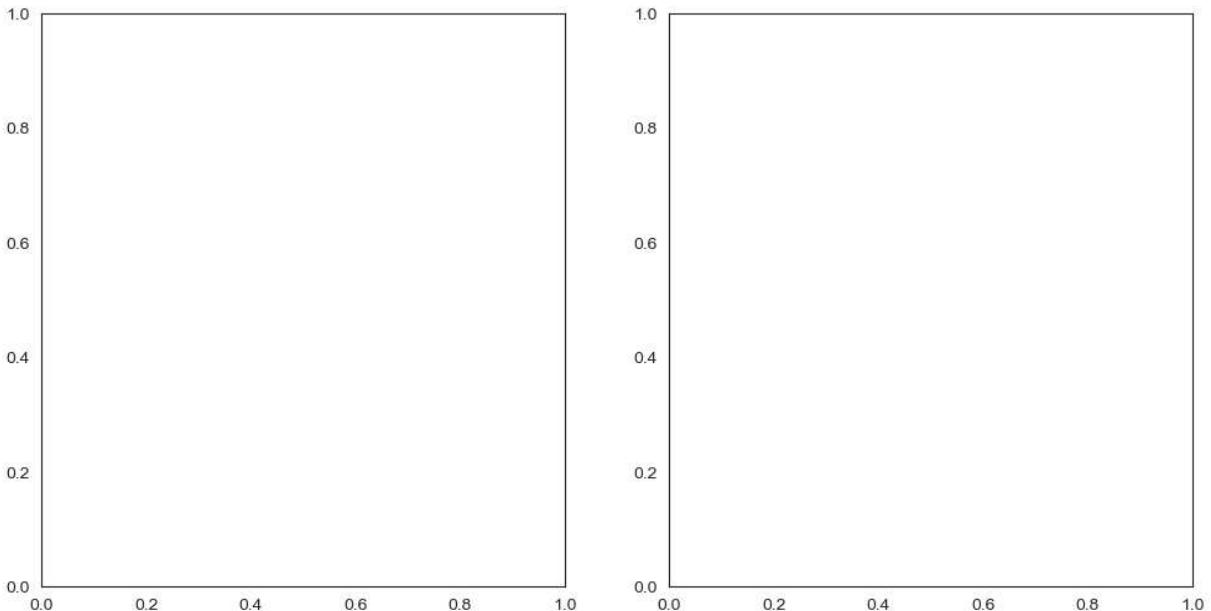


```
In [84]: vis2=sns.lmplot(data=data, x='CriticRating',y='AudienceRating',fit_reg=False,hue="G  
plt.show()
```

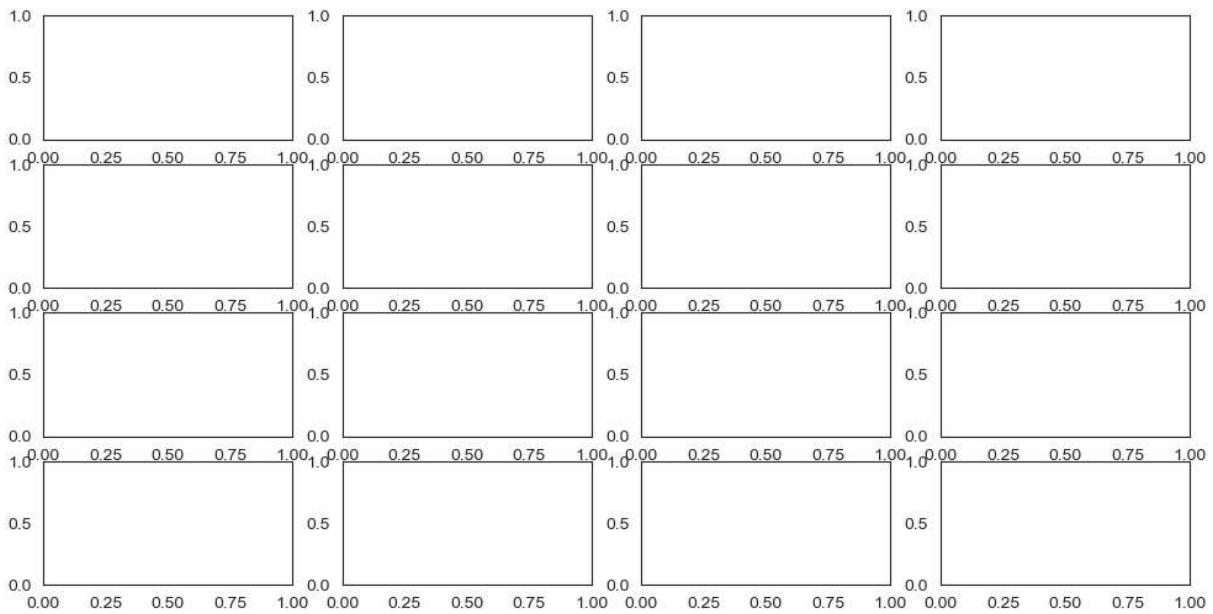


```
In [86]: f,ax = plt.subplots(1,2,figsize=(12,6))
plt.show()
```

<Figure size 640x480 with 0 Axes>



```
In [88]: f,ax = plt.subplots(4,4,figsize=(12,6))
plt.show()
```



In [91]: movies

Out[91]:

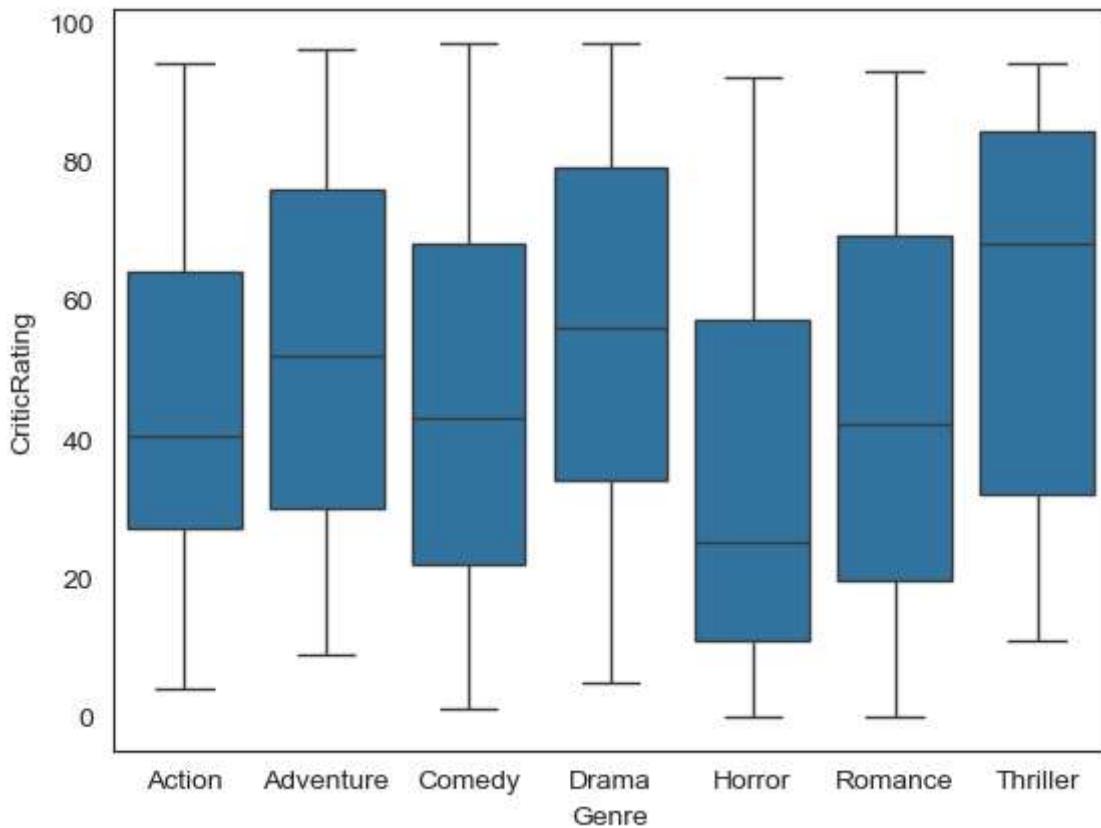
	Film	Genre	CriticRating	AudienceRating	BudgetMillions	Year
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009
...	...	...	...	...	...	...
554	Your Highness	Comedy	26	36	50	2011
555	Youth in Revolt	Comedy	68	52	18	2009
556	Zodiac	Thriller	89	73	65	2007
557	Zombieland	Action	90	87	24	2009
558	Zookeeper	Comedy	14	42	80	2011

559 rows × 6 columns

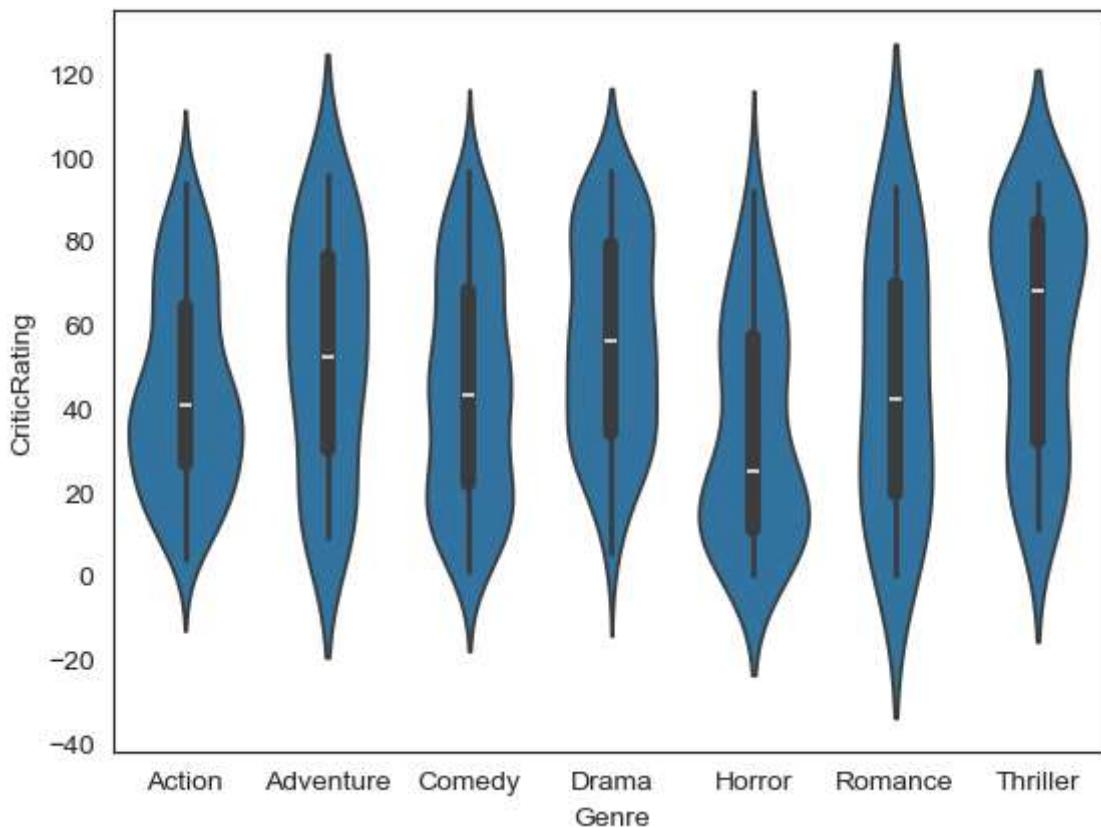
In [92]: sns.boxplot(data=data,x='Genre',y='CriticRating')

Out[92]: <Axes: xlabel='Genre', ylabel='CriticRating'>

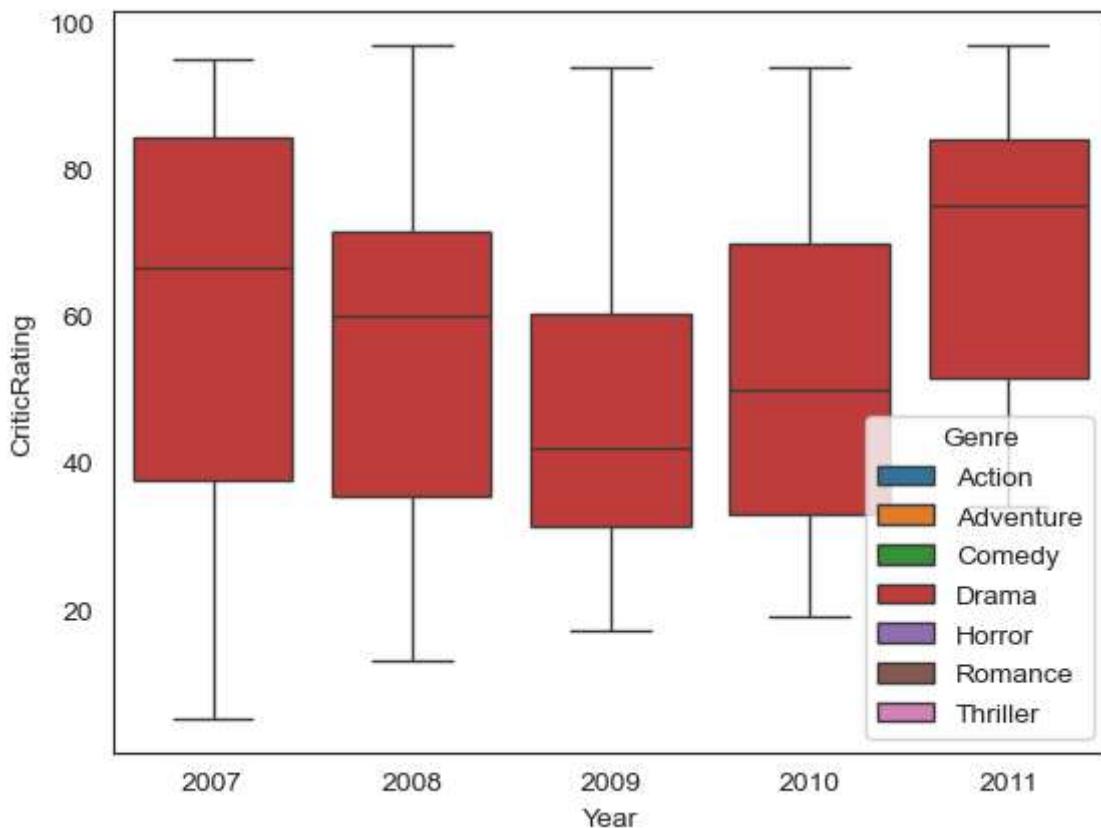
In [93]: plt.show()



```
In [94]: sns.violinplot(data=data,x='Genre',y='CriticRating')
plt.show()
```



```
In [97]: sns.boxplot(data=data[data.Genre=='Drama'],x='Year',y='CriticRating', hue="Genre")
plt.show()
```



```
In [99]: sns.FacetGrid(data, row='Genre', col='Year', hue='Genre')
plt.show()
```

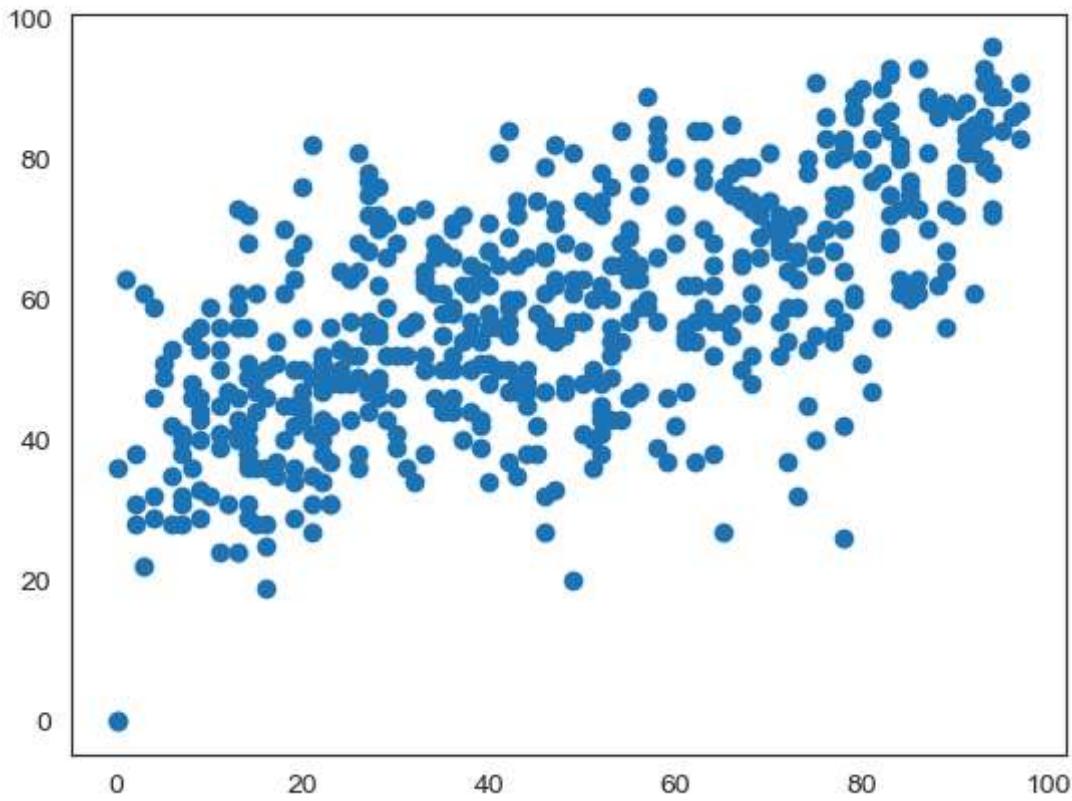
## Seaborn



## Seaborn

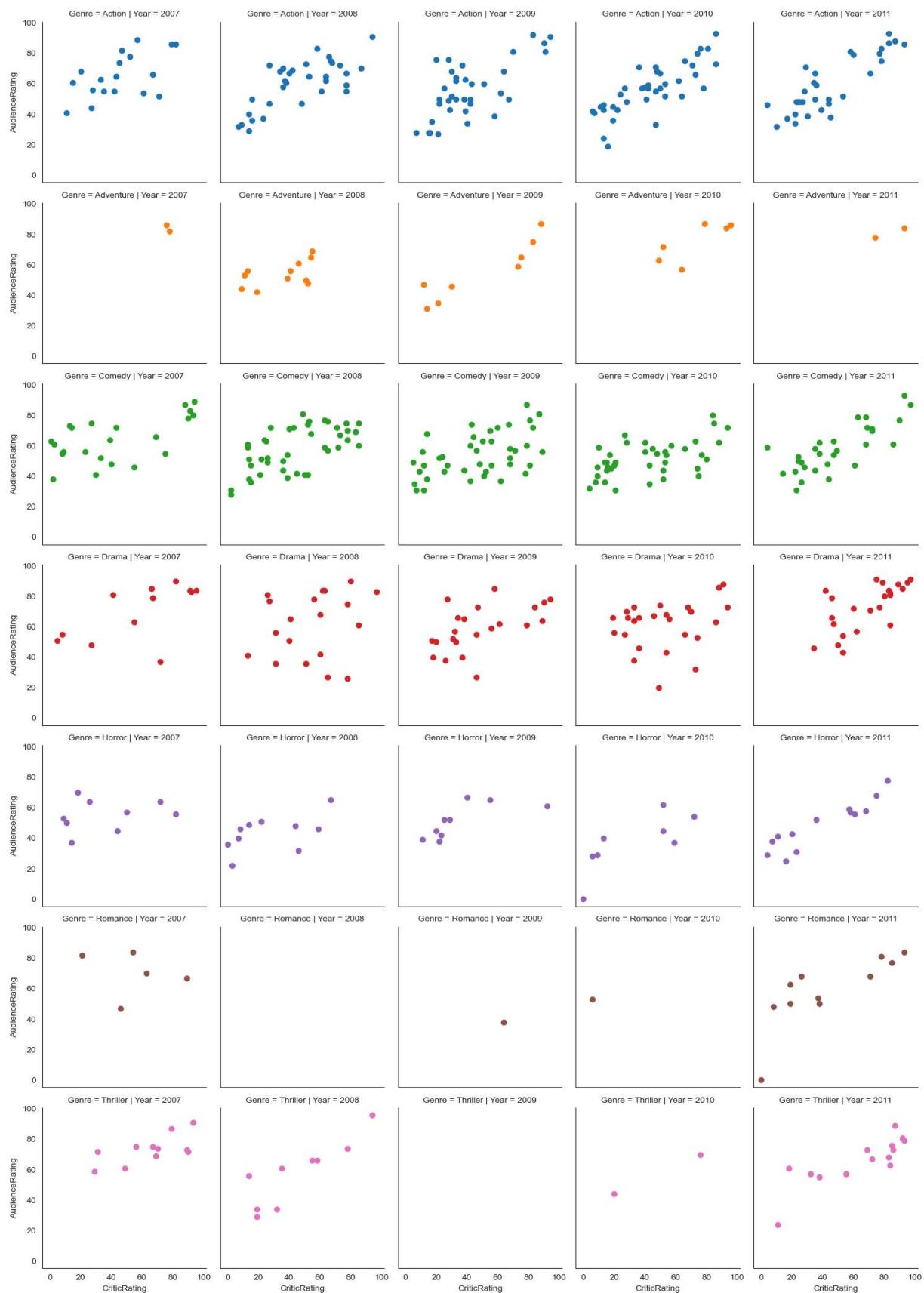


```
In [101]: plt.scatter(data.CriticRating, data.AudienceRating)
plt.show()
```



In [104...]

```
z=sns.FacetGrid(data, row ='Genre', col='Year', hue='Genre')
z=z.map(plt.scatter,'CriticRating','AudienceRating')
plt.show()
```



```
In [105]:  
x=sns.FacetGrid(data,row='Genre',col='Year',hue='Genre')  
x=x.map(plt.hist,'BudgetMillions')  
plt.show()
```



```
In [109]: sns.FacetGrid(data, row='Genre', col='Year', hue='Genre')
kws= dict(s=50, linewidth=0.5, edgecolor='black')
```

```
h=h.map(plt.scatter,'CriticRating','AudienceRating',**kws)  
plt.show()
```



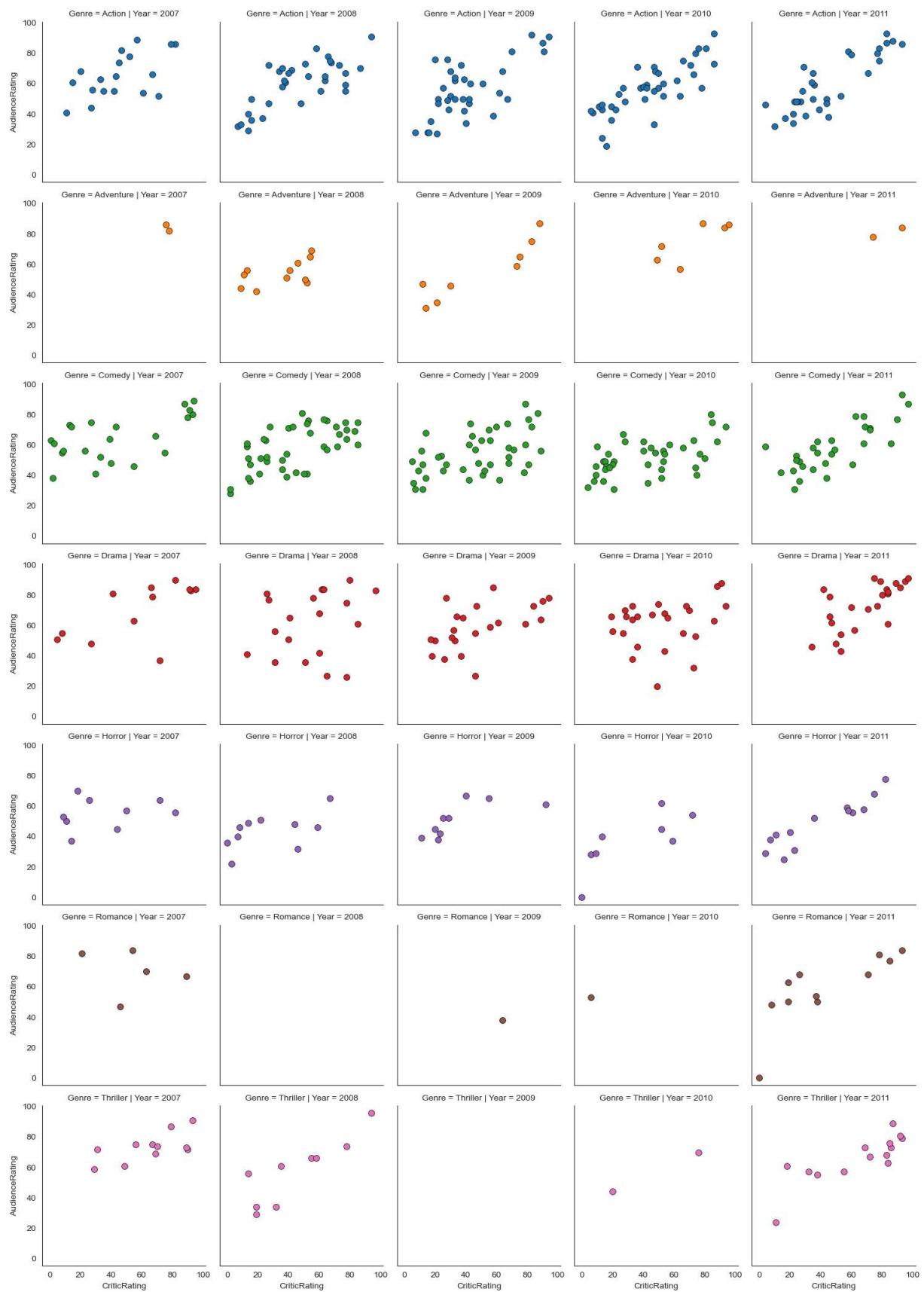
## Seaborn



## Seaborn



## Seaborn



In [ ]: